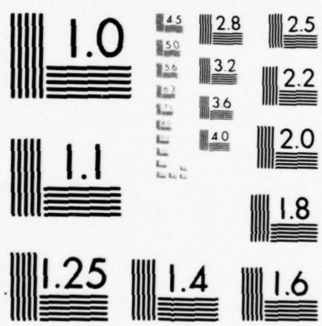
AD-A061 647 CONSTRUCTION ENGINEERING RESEARCH LAB (ARMY) CHAMPAI--ETC F/G 9/2
A SURVEY OF THE PROPERTIES OF COMPUTER COMMUNICATION PROTOCOLS.--ETC(U)
SEP 78 J W LIU, M D MICKUNAS

UNCLASSIFIED CERL-TR-O-1 NL

1 OF 1
AD A
061647

END
DATE
FILMED
7-80
DTIC



MICROCOPY RESOLUTION TEST CHART

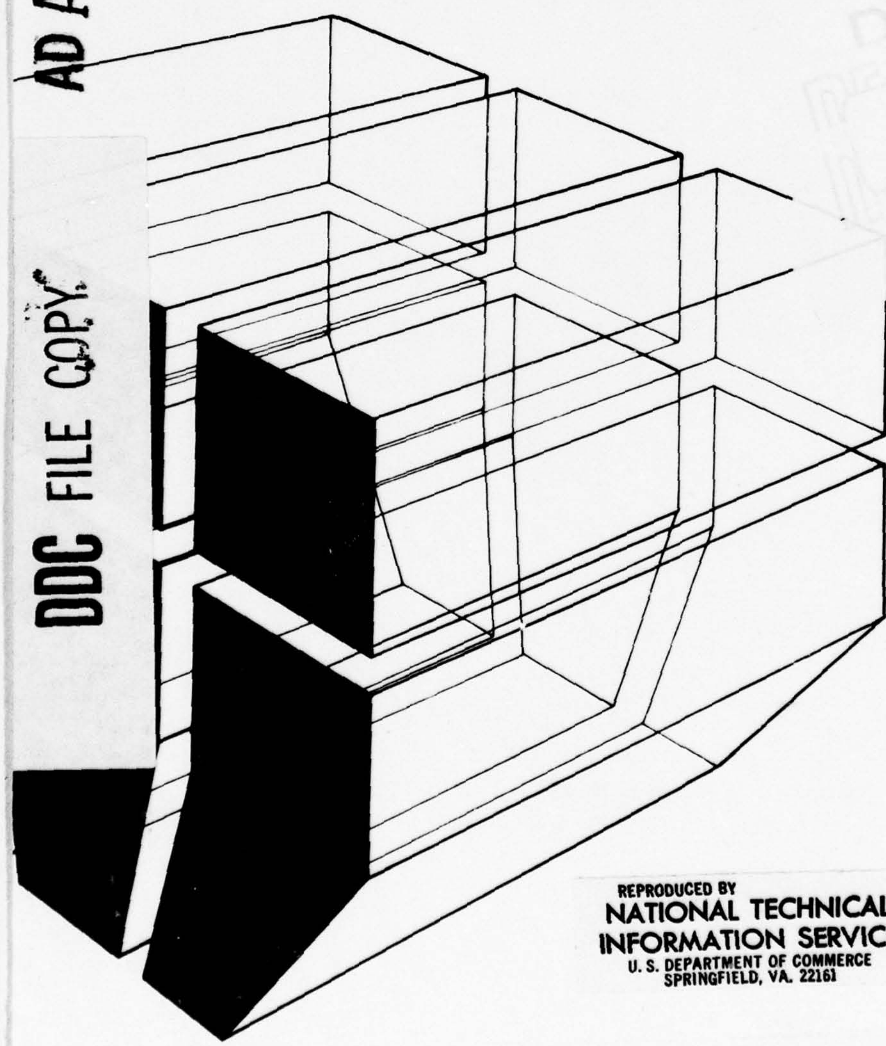
construction
engineering
research
laboratory

TECHNICAL REPORT O-1
September 1978

AD A061 647

A SURVEY OF THE PROPERTIES OF COMPUTER COMMUNICATION PROTOCOLS
VOLUME II: FUTURE DEVELOPMENTS OF COMPUTER NETWORK PROTOCOLS

by
J. W. S. Liu
M. D. Mickunas



CERL

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

REPORT DOCUMENTATION PAGE

READ INSTRUCTIONS BEFORE COMPLETING FORM

14

1. REPORT NUMBER
CERL-TR-0-1

2. GOVT ACCESSION NO.

3. RECIPIENT'S CATALOG NUMBER

6

4. TITLE (and Subtitle)
A SURVEY OF THE PROPERTIES OF COMPUTER COMMUNICATION PROTOCOLS.
VOLUME II. FUTURE DEVELOPMENTS OF COMPUTER NETWORK PROTOCOLS.

9

5. TYPE OF REPORT & PERIOD COVERED
TECHNICAL repts

6. PERFORMING ORG. REPORT NUMBER

10

7. AUTHOR(s)
J. W. S. Liu
M. D. Mickunas

8. CONTRACT OR GRANT NUMBER(s)

9. PERFORMING ORGANIZATION NAME AND ADDRESS
U.S. ARMY ENGINEER
CONSTRUCTION ENGINEERING RESEARCH LABORATORY
P.O. Box 4005, Champaign, IL 61820

16

10. PROGRAM ELEMENT, PROJECT, TASK AREA & WORK UNIT NUMBERS
4A762725AT11-2-209 17 02

11. CONTROLLING OFFICE NAME AND ADDRESS

11

12. REPORT DATE
September 1978

13. NUMBER OF PAGES
34

14. MONITORING AGENCY NAME & ADDRESS(if different from Controlling Office)

15. SECURITY CLASS. (of this report)
Unclassified

15a. DECLASSIFICATION/DOWNGRADING SCHEDULE

16. DISTRIBUTION STATEMENT (of this Report)
Approved for public release; distribution unlimited.

17. DISTRIBUTION STATEMENT (of the abstract entered in Block 20, if different from Report)

18. SUPPLEMENTARY NOTES
Copies are obtainable from National Technical Information Service
Springfield, VA 22151

19. KEY WORDS (Continue on reverse side if necessary and identify by block number)
computer communication protocols
packet switching network
end-to-end protocols

20. ABSTRACT (Continue on reverse side if necessary and identify by block number)

This report is the second part of a two-part study on the properties of computer communication protocols. It is concerned primarily with the future development of protocols, particularly a projection of the state of the art of protocols of different levels. The packet switching network is explained, and the possible future developments in low-level protocols are discussed, including end-to-end protocols and user-level protocols that support functions such as file transfer and remote job entry.

405 279

DD FORM 1 JAN 73 1473 EDITION OF 1 NOV 65 IS OBSOLETE

UNCLASSIFIED

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

78 11 24 074

FOREWORD

This investigation was performed for the Engineer Information and Data Systems Office (EIDSO), Office of the Chief of Engineers (OCE), under Project 4A762725AT11, "Engineering Software Development Methods"; Task 2, "Data and Language Structures"; Work Unit 209, Design of a Machine Protocol Language." The OCE Technical Monitor was Mr. R. McMurrer.

The investigation was performed by the Computer Services Branch (SOC), Support Office (SO), U. S. Army Construction Engineering Research Laboratory (CERL). Personnel directly involved in the study were Dr. J. W. S. Liu and Dr. M. D. Mickunas of the Computer Science Department, University of Illinois, Urbana-Champaign.

Mr. W. Schmidt is Chief of SOC, and Mr. W. Assell is Chief of SO. COL J. E. Hays is Commander and Director of CERL, and Dr. L. R. Shaffer is Technical Director.

Preceding page blank

CONTENTS

A SURVEY OF THE PROPERTIES OF COMPUTER COMMUNICATION PROTOCOLS--VOLUME II: FUTURE DEVELOPMENTS OF COMPUTER NETWORK PROTOCOLS

1 INTRODUCTION

Background

A number of large-scale computer networks that use packet switching networks as communication media between computers and data terminals have been developed, planned, and implemented in recent years. To insure meaningful, reliable information exchange between the computers and terminals, many protocols have been designed and implemented.* In order to develop a useful scheme for verifying, analyzing, and implementing communication protocols, it is necessary to understand both the current and future requirements and capabilities of communication protocols. Current relevant issues in protocol design (such as flow control, synchronization, resiliency, etc.) have been discussed in Volume I of this report.[1]

Purpose

The purpose of this report is to discuss the state of the art of protocols of different levels and possible future developments of protocols.

Outline of Report

Chapter 2 contains a general description of packet switching networks, the type of communication subnetworks of primary concern in this report. To avoid possible ambiguity in subsequent discussions, Chapter 2 also summarizes the functions of all protocols necessary to achieve reliable and smooth information exchange in a communication network. Chapter 3 discusses the possible future developments in low-level

---

[1] A. Itzkowitz, *A Survey of the Properties of Computer Communication Protocols, Volume I, The Function, Properties, Specification, and Analysis Methods of Computer Communication Protocols*, Technical Report O-1 (Construction Engineering Research Laboratory, September 1978).

*General discussions on computer networks and communication protocols can be found in articles listed in the uncited references. Citations listed here are primarily survey articles, each containing more complete citations on these subjects.

protocols (up to and including end-to-end protocols). Chapter 4 discusses user-level protocols to support common requirements such as terminal supports, file transfer, and remote job entry.

2 GENERAL DISCUSSION

In packet switching networks, messages to and from user processes are broken up into small segments known as packets. Packets are individually routed through the communication network, rather than having a physical circuit setup linking the communicating processes and reserved for the duration of the conversation. Each node along the route transversed by a packet *stores and forwards* the packet to the next node. The transmission of a message consisting of one or more packets is completed when the receiving process acknowledges the correct reception of all packets in the message. Figure 1 shows a packet switching network. Because it combines reliable, low-delay communication with high-circuit utilization, packet switching technology is used in most contemporary networks which can tolerate only short communication delay.

To insure smooth and error-free information exchange between user processes, a combination of software and hardware interfaces must be introduced at various points in the communication network and between host computers, data terminals, and the network. These interfaces enforce the procedure rules which specify the responsibilities of communicating processes in error detection and recovery, information flow control, synchronization, etc. These procedure rules are commonly referred to as *protocols*.

Conceptually, protocols may be grouped vertically so that each group can be viewed as a functionally isolated layer. Different layers of protocols govern different levels of network interactions. One of the best known examples of layered protocols is the ARPANET protocols shown in Figure 2. At the lowest level, the communication subnetwork protocols* guarantee reliable transmission of data among the packet switching nodes. This level of protocol specifies error control and routing procedures to provide low-delay delivery of the messages. The interface between host computers to the communication subnetwork is at a higher level. This level specifies formats and timing of messages transmitted between the host computers and data terminals to the packet switching network. The host-to-host level protocol deals with rules for establishing and maintaining a virtual circuit for each pair of communicating processes. The telecommunication network protocol provides mechanisms by which user processes at remote hosts or data terminals can gain access to a local interactive system. Finally, at the user level,

---

*In most literature on ARPANET, communication subnetwork protocols are referred to as IMP-IMP protocols.

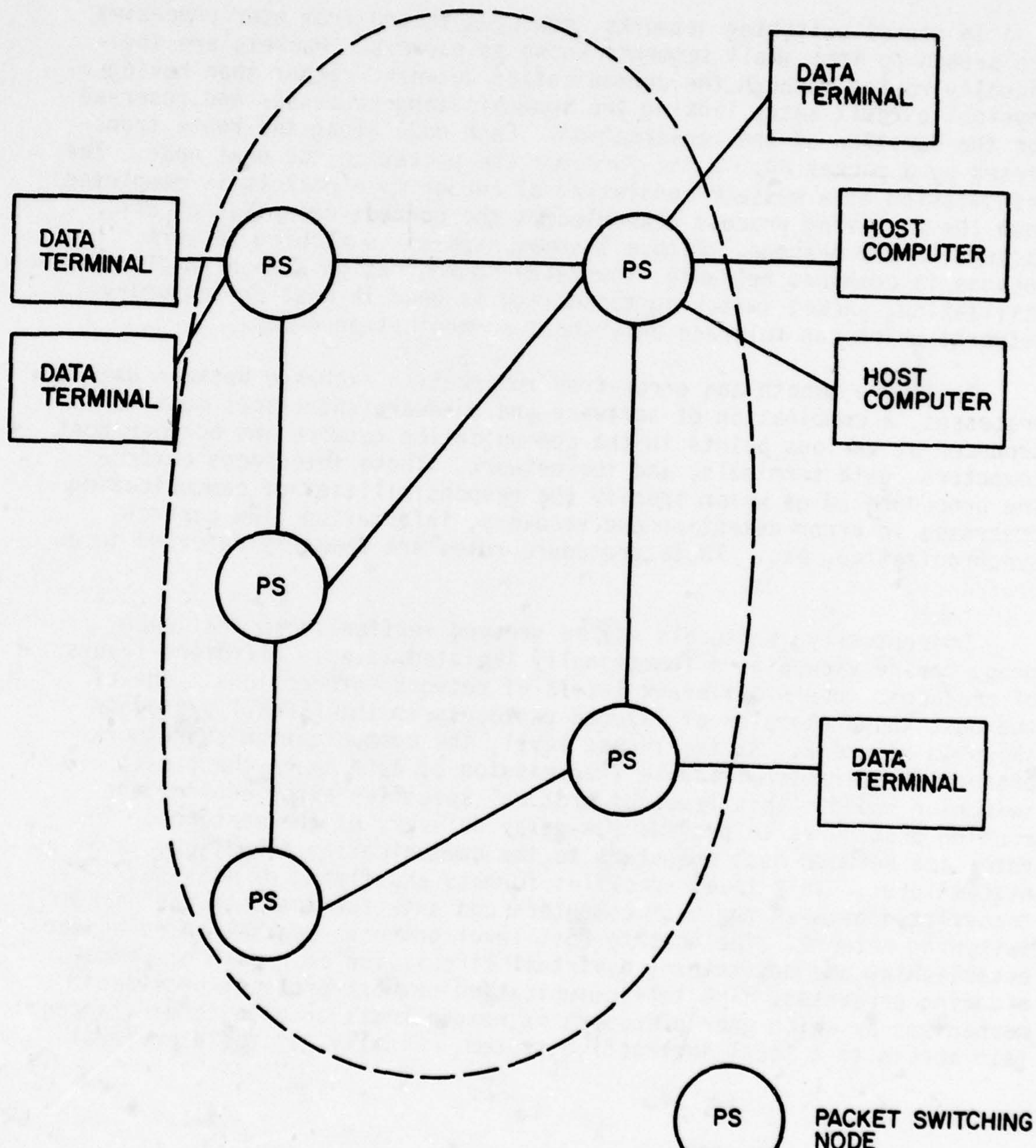


Figure 1. Typical packet switching network.

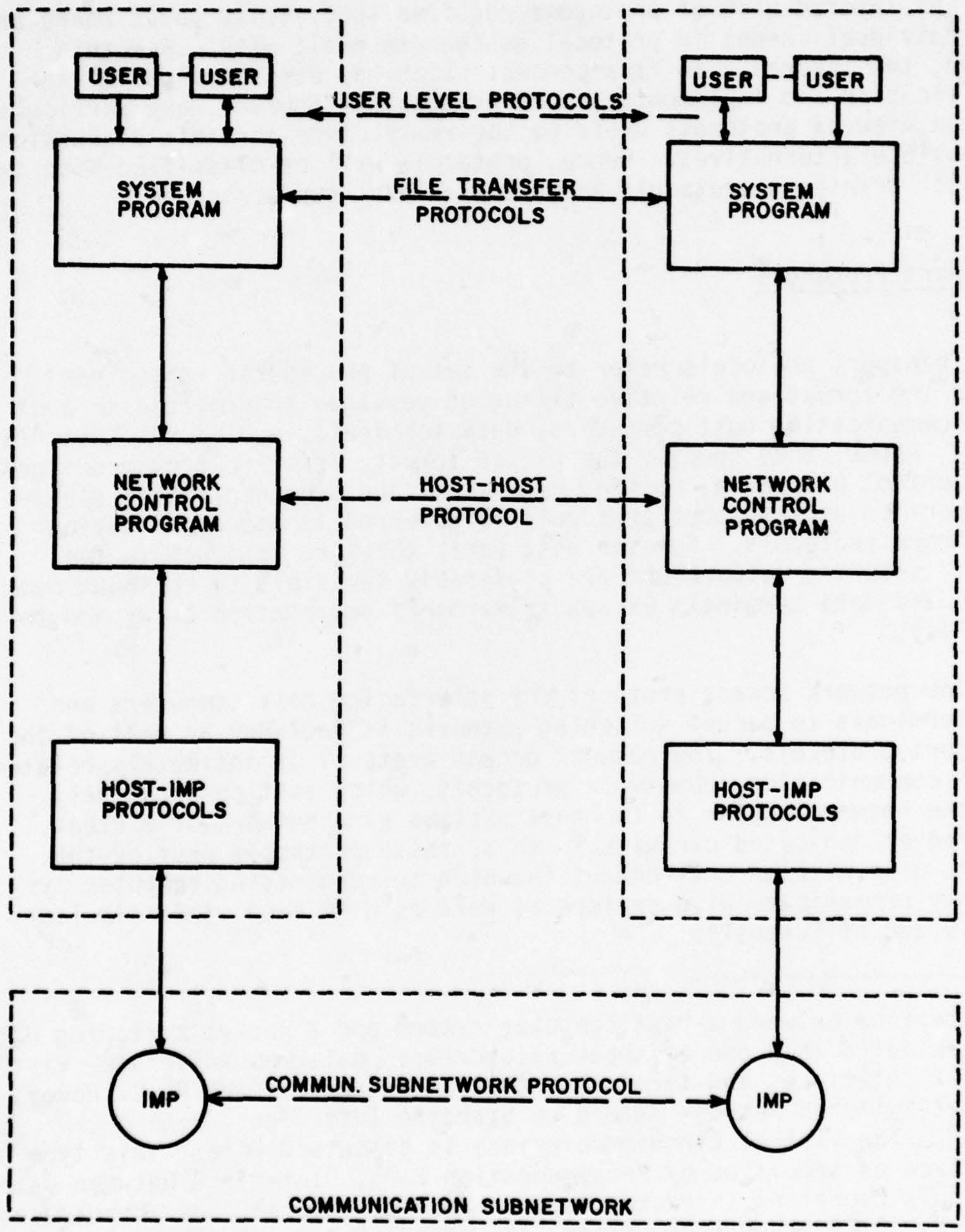


Figure 2. Layered view of ARPA network protocols.

protocols such as file transfer protocols, procedure call protocols, and remote job entry protocols, allow efficient use of resources in the network.

The layered view of protocols requires specificity about where and how individual issues in protocol design are dealt with. For this reason, the layered view is a concept which has been very useful in specification and implementation of protocols. However, any particular layered view of protocols would be too restrictive for this discussion of possible alternatives. Hence, protocols will be classified into two groups: transport protocols and process-level protocols.

Transport Protocols

Transport protocols refer to the set of procedural rules that govern the format and relative timing of messages transmitted to and from communicating host computers, data terminals, and packet switching nodes. Hence, they specify the packet format, error control procedure, flow control procedure, routing algorithm, and synchronization scheme. This portion of transport protocols is referred to as communication subnetwork protocols. For the most part, they are internal to the packet switching network and are preferably invisible to the host computers and data terminals except in terms of propagation delay and reliability.

The network access protocol for interfacing host computers and data terminals to packet switching networks is included as part of the transport protocols. The network access protocol is intimately related to the communication subnetwork protocols, which must cooperatively make the network appear to the host systems as a network of virtual switched or dedicated circuits.* Thus, these protocols provide the network user with an environment in which teleprocessing computer systems may communicate with devices as well as with each other simultaneously and efficiently.

---

*An interface between a host computer system and a packet switching data network falls into one of three categories: datagram interface, virtual circuit interface, and terminal emulation interface (see R. B. Hovey, "Packet-Switched Network Agreed on Standard Interface," *Data Comm* [1976]). The virtual circuit interface is discussed here. This type of interface as specified by Recommendation X.25, "Interface between Data Terminals Operating in Packet Mode on Public Networks," has been recently voted by CCITT study group VII as the standard network access interface. It has been adopted by Telenet Communication Corporation, Bell in Canada, and the telecommunications operating public packet switching works in France and Britain. In Chapter 3, the strengths and weaknesses of this type of interface are discussed, and possible future modifications are described.

Process-Level Protocols

Process-level protocols are those procedural rules which specify mechanisms by which connections are established and maintained, interactive host systems are accessed via remote terminals, files are transferred among host computers and data terminals, and batch jobs are entered at remote sites. (These protocols are referred to in ARPA network literature as the initial connection protocol and function-oriented protocols[2].) Most of the issues dealt with in these protocols are not unique to teleprocessing computer systems. Most of these procedural rules are required in any system in which a computer must support many terminals and communicate with processes residing in other computers.

---

2 S. D. Crocker, J. Heafner, R. Metcalfe, J. Postel, "Function-Oriented Protocols for ARPA Computer Network," *AFIPS 1972 SJCC Proc.*, 40 (1972), pp 271-279.

3 TRANSPORT PROTOCOLS

This chapter discusses the state of the art of transport protocols and foreseeable future developments in these protocols. Since the advent of computer networks, a great deal has been learned about the relative merits of alternative designs of transport protocols and about the performance of protocols now being used. To date, although many aspects still need improvement, the development of communication subnetwork protocols has reached a reasonably mature stage. Widely accepted designs and adopted or de facto standards exist for most mechanisms, and are summarized briefly in the following sections.

In the area of network access protocols, there are still differing views on the division of burden between the host computers (and/or data terminals) and the communication subnetwork to insure reliable and smooth delivery of data. Different network access interfaces have been proposed by various interest groups. These alternatives and their implications on more practical issues are discussed in more detail below.

Communication Subnetwork Protocols

The development of communication subnetwork protocols has benefited greatly by the knowledge of data communication technology. Characteristics of currently used communication media (including the satellite channel) are understood well enough to allow sound design of error control procedures and bit-level synchronization schemes. For example, since errors in telephone data communication channels are known to occur in bursts, continuous automatic-repeat-request (ARQ) systems offer an effective error control mechanism at acceptably high effective data rates in most data communication networks.[3] Hence, the error control scheme of using a cyclic redundant code together with a positive acknowledgement and negative timeout scheme has been widely used. (By using 16 redundant bits per packet, the probability of undetected error is approximately 10^{-8} per packet for most packet sizes used presently.) The only question remaining is whether node-to-node ARQ (as in the ARPANET IMP/IMP protocol) is really required, given the availability of acknowledgement from destination node and/or host computer (for example, in the form of Ready for Next Message, RFNM, facility in the present ARPANET Host/IMP protocol. However, given the proper network access protocols, the exact implementation of any error control mechanism should be invisible to network users.

---

[3] H. O. Burton and D. O. Sullivan, "Errors and Error Control," *Proceedings of IEEE*, 60 (November 1972).

The design of packet formats, receiver addressing schemes, and packet level synchronization has also been aided by experience gained in the design of data link control procedures in local teleprocessing systems. Since individual packet switching networks may differ in their implementations, these differences may pose certain difficulties to designers and implementers of internetwork protocols, but should impose no real inconvenience to the network user.

Flow control is a somewhat more complex problem. Its object is to achieve a globally smooth flow pattern at a near-maximized traffic volume, and its solution also dictates extensively the requirements of the network access interface and, in many cases, network control programs and operating systems in the host. Alternative solutions include requiring preallocation of buffer spaces in the receiving host computer and packet switching node before allowing transmission of a packet; using permits at source nodes to minimize congestion; discarding of messages which arrive when there is not enough buffer space in the receiving nodes; relieving the receiving packet switching node of the responsibility of reassembling the packets into messages, detecting duplicates, etc.[4] These solutions have been shown through both analysis and experimental observation to exhibit undesirable characteristics. For example, in ARPANET, flow control is done at the host-to-host level by requiring the receiving host computer to allocate buffer space for each virtual circuit and to notify the sending host of the amount. As the messages are transmitted, the sending host must keep track of the remaining amount of free buffer space. Messages are allowed to enter the source IMP only when they can fit in this free buffer area. Hence, to maintain throughput, the receiving host must periodically send allocations to the sender. A recent study has shown that host systems generally allocate too little buffer space;[5] consequently, almost 80 percent of all control messages transmitted over ARPANET are allocate commands. This overhead represents more than 40 percent of the packets transmitted. Together with the overheads used in addressing, synchronization, error control, etc., the data rate for process to process communication will be limited eventually to 23 percent of the data rate of the communication link. A more serious problem caused by this flow control scheme is the loss of synchronization between the receiving and sending hosts when an allocation command is lost or duplicated. Alternatively, the source may be allowed to transmit at any time and the receiver may be permitted to treat as erroneous any message that it cannot handle due to buffer overflow. This scheme will result in extra retransmission which may create congestion.

---

4 V. G. Cerf, *An Assessment of ARPANET Protocols*, RFC 635, NIC 30469 (Network Information Center, Stanford Research Institute, 1974).

5 L. Kleinrock, W. E. Naylor, and H. Opderbeck, "A Study of Line Overhead in ARPANET, " *Comm. of ACM* (1976), pp 3-12.

There are many proposals for modifying existing communication subnetwork protocols to increase network throughput via better and more robust flow control schemes.[6] The evaluations of these new proposals along with the issue of message level synchronization are discussed in the following section.

Network Access Protocols

In order that user process communication be conducted without errors and undue delays, communication subnetwork protocols and network access protocols must cooperatively provide all the necessary mechanisms to detect bit-level and message-level errors, to detect loss and duplication of messages, to prevent any host computer or data terminal from overloading the interface link of another host computer, etc. Together with other hardware/software interfaces in the host computers and/or data terminals, they constitute what is usually known as end-to-end (or host-host, or second-level) protocols.

The Development of Existing End-to-End Protocols

The best known end-to-end protocol is the ARPANET host-host protocol. Its design is based on the assumption that interprocess communication will be based on exchange of a sequence of messages rather than on one message. Therefore, for each pair of participating processes, a virtual simplex link (connection) is established so that an output port of one process is the input port of the other. As shown in Figure 3, the Network Control Programs (network access interfaces) implement control functions required for establishing and breaking connections and for controlling data flow over the connections. The control messages are exchanged between host computers over a control link (link 0) established and maintained between each pair of host computers in the network. When processes in different hosts want to communicate with each other, control messages are exchanged as specified by initial connection protocols to set up a connection (or a pair of connections). When a connection is set up, the receiving host allocates a link number. The sending host specifies the byte size to be used in communication over the link. The link number is then used with the host number to identify the connection and for host-host flow control. Since the control link is always open for subsequent exchanges of control commands between hosts, socket number is used only for breaking the connection. The Network Control Program in the sending host has the responsibility of segmenting interprocess communication into network messages. Similarly,

---

6 *An Assessment of ARPANET Protocols.*

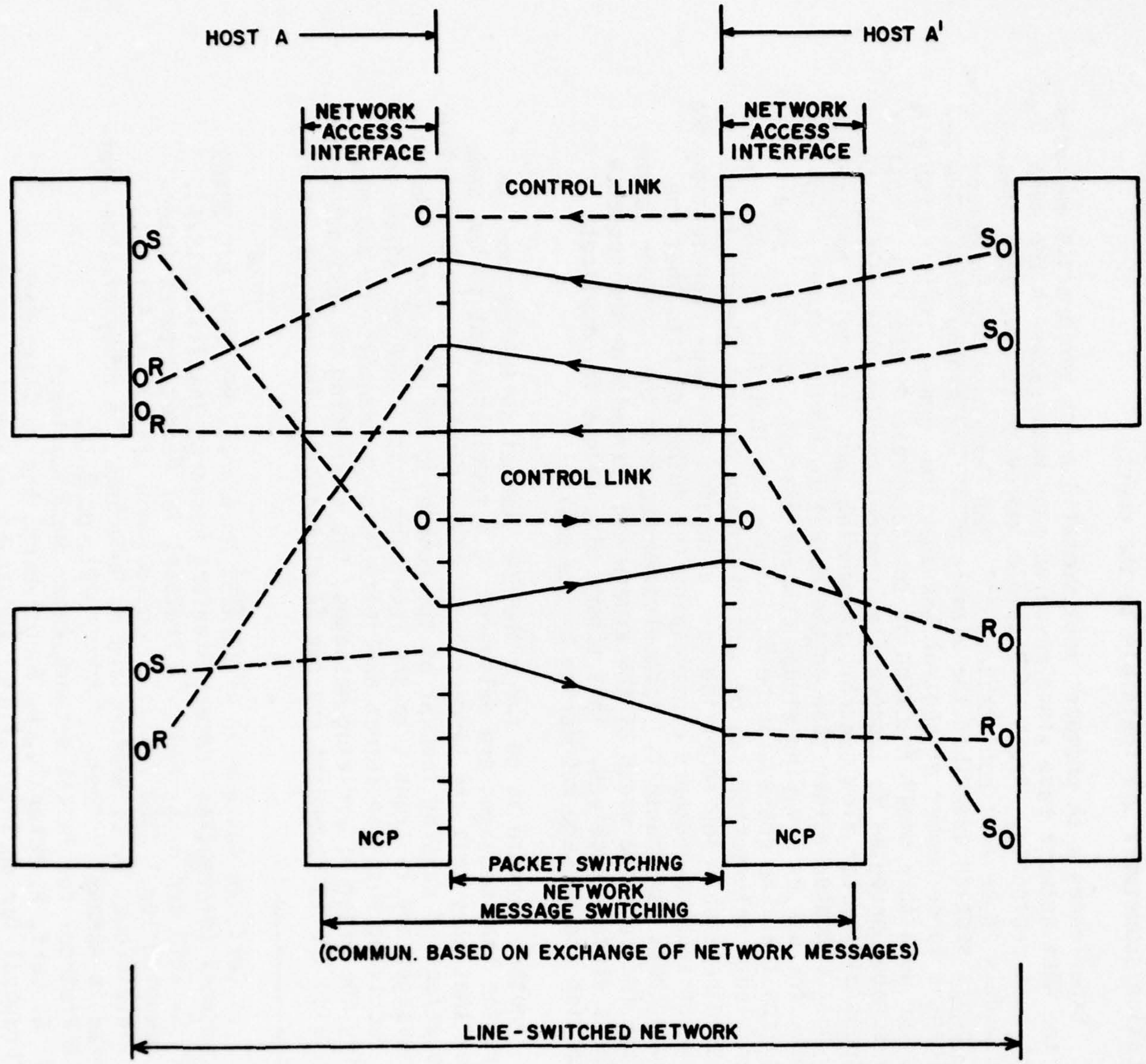


Figure 3. ARPANET host-host protocol.

the Network Control Program in the receiving host concatenates successive messages from the network into a single transmission and delivers it to the receiving process. Therefore, to the user processes, the network appears to be a line-switched network where communication links are set up by calling the receiver. The message level exchanges between the network access interfaces and the packet switching within the communication subnetwork are invisible to the users.

Experiences with protocol implementations and performance analysis in the ARPA network have since revealed many weaknesses in the ARPA host-host protocols. Areas of possible improvements have been discussed widely in several publications.[7-9] The Communication Subnetwork Protocols section described the example of a flow control procedure requiring a large number of control messages and, hence, limiting the effective data to a small fraction of the data rate of the link. Most of the schemes proposed to improve flow control require that more of the responsibility for flow control, sequencing, and addressing be placed on the host computer rather than on the packet switching network. For example, the use of a moving window concept discussed by Cerf, et al.,[10,11] requires that the connection be full duplex so that flow control and acknowledge information can be sent piggyback on data flowing in the reverse direction. The receiving host allocates a window representing the range of sequence numbers constraining the number of bits that the sending host may transmit. Acknowledgements from the receiver to the sender indicate the width of the window and acknowledge the sequence numbers already received. This scheme also allows for duplicate detection and message reordering by the hosts.

Another example is to force the IMP (packet switching node) to guarantee that messages are delivered to a receiving host in the same order that they left the sending host. This service has the undesirable side effect of causing lockups at the receiving IMP when large numbers of messages are in transit to the receiving host. However, since only the receiving host can detect duplicate message transmission and must verify the order of arriving messages, the sequencing of messages by the receiving IMP is redundant and can be eliminated. Furthermore, as long

---

7 V. G. Cerf, *An Assessment of ARPANET Protocols*, RFC 635, NIC 30469 (Network Information Center, Stanford Research Institute, 1974).

8 V. G. Cerf and R. E. Kahn, "A Protocol for Packet Network Intercommunication," *IEEE Trans. Communications* (1974), pp 637-648.

9 L. Kleinrock, W. E. Naylor, and H. Opderbeck, "A Study of Line Overhead in ARPANET," *Comm. of ACM* (1976), pp 3-12.

10 "A Protocol for Packet Network Intercommunication."

11 V. G. Cerf, A. McKenzie, R. Scantleburg, and H. Zimmerman, "Proposal for an International End-to-End Protocol," *Computer Comm. Review*, Vol 6, No. 1 (1976).

as the receiving host has the responsibility of reordering a message, there is no reason not to let it reorder packets as well. Thus, multi-packet messages can also be eliminated, allowing the sending host to transmit a large number of single-packet messages instead. Since it is also necessary to introduce some kind of end-to-end positive acknowledgement along with host retransmissions, the positive acknowledgement currently sent by the receiving IMP in the form of RFNM is also redundant and can be eliminated.

These arguments, together with experience gained with the French CYCLADE network and British NPL network, serve as bases to the end-to-end protocol proposed by the International Federation of Information Processing Societies working group TC6.1.[12] The international end-to-end protocol proposed by group TC6.1 involves the concept of a virtual host, i.e., a collection of resources which appears as a single entity to the packet switching subnetwork. Packets are sent by a source virtual host to a destination virtual host. Each virtual host contains a transport station (TS) which is responsible for multiplexing the interface to the communication subnetwork. Within a particular virtual host (identified by its TS) are a number of ports (PT). The combination of TS and PT identifiers (a 16-bit identifier with the high-order bits identifying the TS and the low-order bits identifying the PT) provides a network-wide name space. A subset of the port identifiers within each of the TSs is devoted to standard services which are provided by that host. Before PTs can communicate, they must become associated (i.e., they must set up an association), using an Initial Connection Protocol (ICP). The association is full duplex and is completely identified by the pair of end addresses. The protocol provides for the transport of letters (LT) from one port to another. Fragmentation of letters into packets is performed by the TS, and reassembly is done by the destination TS. LTs have a 16-bit reference number (MY-REF). LTs may be broken into fragments (FR) (otherwise known as packets), each of which has its own number (FR-NB) and carries the MY-REF. The final FR of the LT carries an end-of-letter (EOL) flag. Associations may operate simultaneously in liaison and lettergram modes. Liaison mode is used primarily to initialize a connection, i.e., to make sure that both ends of the liaison are active, and to mediate an agreement on the set of services which will be used, as well as to initialize parameters coherently. In lettergram mode, letters are sent independently with the possibility of acknowledgement requested by the sender. For error control, an end-to-end checksum for the entire LT is placed in the final FR. The protocol recognizes that it may be necessary to transmit messages between networks of different packet sizes. Between subnetworks, "gateways" will

---

[12] V.G. Cerf, A. McKenzie, R. Scantleburg, and H. Zimmerman, "Proposals for an International End-to-End Protocol," *Computer Comm. Review*, Vol 6, No. 1 (1976).

be provided to fragment packets and to supply proper headers throughout (including transport station headers within the new fragments). In addition, the TC6.1 recommendation contains some suggestions for generic primitives for interface to the network-access-method. These include primitives to activate a port, to receive a letter from any distant port in lettergram mode, to send a letter in lettergram mode, to de-activate a port, to initialize a liaison, to receive a letter in liaison mode, to send a letter in liaison mode, and to terminate the liaison. The proposal also takes into account plans for future networks and interconnection of different networks. The protocol provides for full duplex connections and permits control information to be carried piggyback on data flowing in the reverse direction. It uses the moving window concept for both flow control and lost (or duplicate) message detection and is self-synchronizing to some extent. Since it is a sequencing positive Acknowledgement/Retransmission protocol, it is believed to be robust and has the additional advantage of requiring only a minimal number of assumptions about its operating networks.

From the user processes point of view, the proposed international end-to-end protocol (see Figure 4) is based on message switching discipline. The network access interface (referred to as transport station or transmission control program) can be considered by the user processes as extending the packet switching network to the port level. When two ports want to communicate, their intent is established according to a port-to-port protocol and thus becomes associated; hence, an association can be viewed logically as a full duplex link identified by the complete addresses (network address/host address/transport station/port of the user process) of the source and destination. To send a message after having learned the globally unique identification of the intended receiver, the user process adds control information to its text that contains the complete address of the receiver. Along with other network control information,* the message is sent to the transport station in the form of a completely self-contained *letter*. The transport station then fragments the letter into network packets (*datagrams*, completely self-contained) and delivers them to the packet switching network. The network handles datagrams independently so that each is delivered to the receiving transport station with minimum delay; the receiving host thus has the tasks of detecting lost data, duplication, reordering messages, eliminating traffic congestion, etc. The receiving transport station accepts the packets, reconstructs the messages, and delivers them to the receiving process.

*Also contained in a transit control block are the following: information on source address, sequence number to be used for the next packet to be sent from this transport station, the present size of the process transmit buffer, the address of the next position in the buffer at which the process can place new data for transmission, the number of times the transport station has not transmitted the data, the location of data still unacknowledged by the receiving transport station, etc.

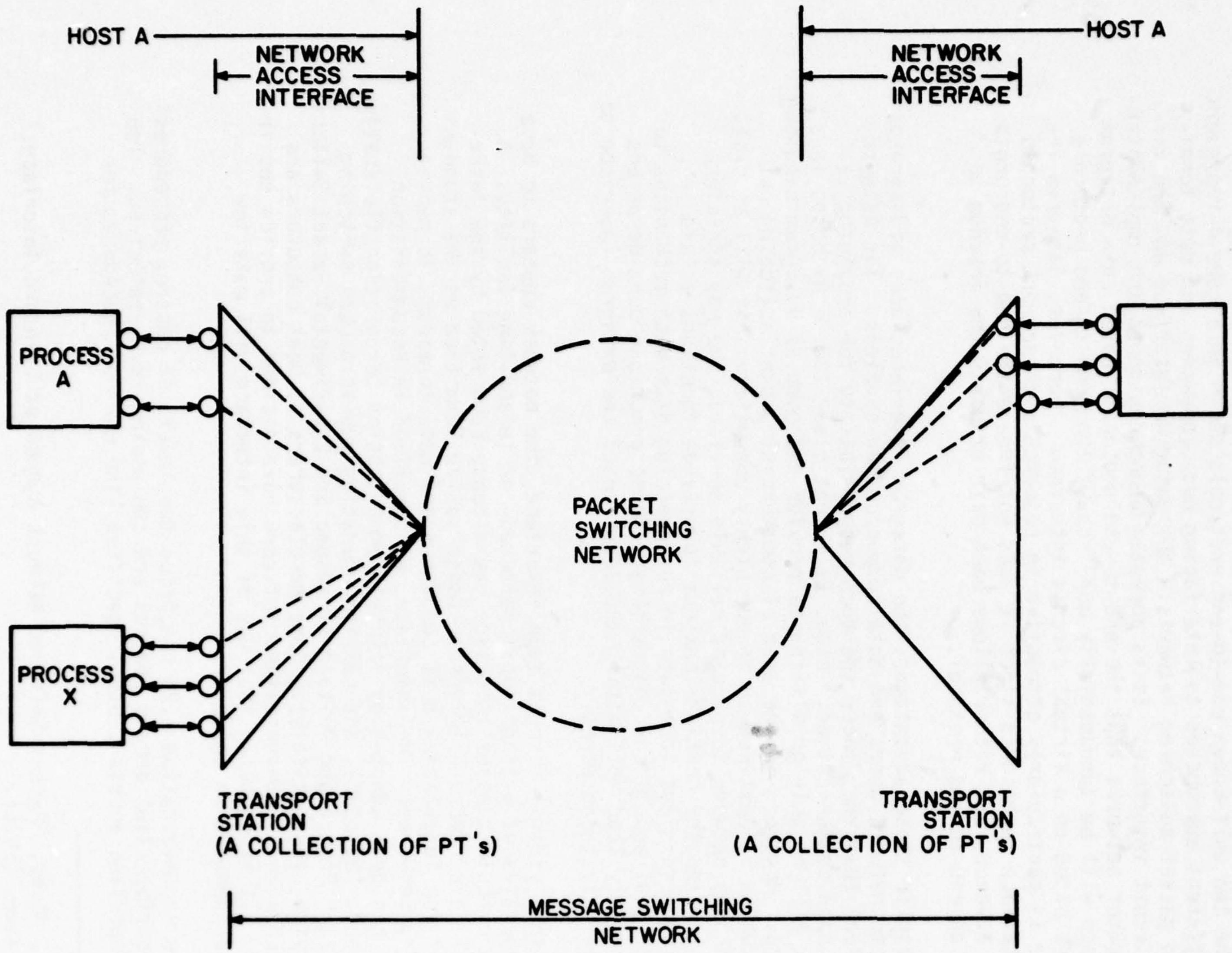


Figure 4. Proposed international end-to-end protocol.

The two well-known end-to-end protocols shown in Figure 4 represent two different approaches to interfacing host computers and data terminals to packet switching networks: the datagram interface and the virtual circuit interface. It is accepted among designers and implementers of computer networks that the end-to-end protocols based on a datagram interface will be fundamentally more robust than end-to-end protocols that are based on a virtual circuit interface. Moreover, datagram interface is particularly attractive in research with network protocols and interface techniques since it does not freeze the end-to-end protocol in advance, but rather allows each pair of hosts the freedom to choose an end-to-end protocol.

Despite these advantages, the datagram interface lacks widespread appeal to network users and data communication carriers. The network users feel that they incur too much responsibility for providing reliable data communication. Also, it is not clear that many hosts in a network will be able or willing to provide the complex datagram handling software. However, in the area of commmercial packet switching networks, which is destined to become highly competitive, it will be critical for the network to provide reliable service. One may therefore expect to find the carriers anxious to provide functions to aid in detecting errors and lost data in resequencing data with mechanisms to prevent local and global congestions. These practical considerations will probably lead to limiting the adoption of the datagram interface to isolated private networks.

With a virtual circuit type interface, the network appears to have the properties of a traditional switched- or leased-line facility. A virtual circuit type of interface has already been voted by the International Telegraph and Telephone Consultative Committee as the standard protocol for interfacing host computers and data terminals to packet switching networks. The interface, as specified in recommendation X.25,[13] has been adopted by Telenet Communication Corporation (Telenet), Bell Canada (Datapac), and carries operating public packet switching networks in France and Britain (Tramspac and Experimental Packet Switching Service, respectively). Some manufacturers of host computers and some suppliers of communication software have decided to provide the interface. It is expected that use of this interface will make few technical demands on the user.

The recommendation X.25 specifies two levels of control procedures: the subscriber-link access control and the packet-level interface. The subscriber-link access control specifies line control procedures for

---

[13] R. B. Hovey, "Packet-Switched Network Agreed on Standard Interface," *Data Comm* (1976).

connecting a host computer or data terminal to the network. Information transferred over the full duplex point-to-point subscriber access link is formatted in the form of a transmission block called a *frame*. The supporting hardware may employ either the traditional byte-oriented data link control mechanism (BSC, Binary Synchronous Communication) or bit-oriented data link control mechanism (SDLC, Synchronous Data Link Control). At the frame level, the procedure uses the principles and terminology of the High Level Data Link Control procedure (HDLC) specified by the International Organization for Standardization. It is primarily responsible for transmission error detection and recovery; thus, it insures that information transferred to and from the network is free of error.

The packet-level interface uses HDLC-like methods and formats for sequencing and flow control, but the HDLC grammar is extended to provide features for establishing virtual circuits. The packet-level interface specifies the control procedures for exchange of call control information and user data.

Practical Impact of the Standard Interface

The adoption of recommendation X.25 as international standard network access interface has many practical consequences. It is expected that standard software interface will be provided by manufacturers of host computers, communication front-ends, and data terminal equipment, since such an option will enhance their standard products. Thus, in the future, packet switching networks may be used for both domestic and international data communication and will make few technical demands on the users. The standardization of network access interface will undoubtedly provide incentive for developing communication products that will implement interdevice protocols in hardware and firmwire.

The shortcoming of the virtual circuit of interface as defined by recommendation X.25 is that the interface is designed to follow a transmission procedure that does not conform to HDLC. Redundant functions provided at the packet and data link control levels in X.25 introduce incompatibility between data terminal equipments with X.25 interface and data terminal equipments with HDLC interface. Since a data terminal equipment with an X.25 interface cannot communicate with a data terminal equipment with HDLC interface, special interfaces will be needed for different configurations. Furthermore, data terminal equipments with X.25 interface cannot be accessed through a switched circuit network.

4 USER ACCESS PROTOCOLS

User access protocols specifically mean the telecommunication network protocols (or interactive terminal protocols) and remote job entry protocols. These facilities are provided by most networks.

A telecommunication network protocol provides an interface between a terminal device and a terminal-oriented process. It allows a user who is at an interactive terminal and who is connected to his/her local host computer (or terminal handler) to control a process in a remote host. The protocol specifies the manner in which characters from the user's terminal are mapped into network-wide transmission format and, at the remote site, are mapped into remote host computer characters. Other necessary features of this protocol include procedures that will enable a user to establish connection and login and that will support echoing and attention handling.

Remote job entry protocols support the batch processing function. Such protocols provide user access to remote host computers for relatively compute-intensive applications. In the layered view of protocols, a remote job entry protocol is placed at a higher layer than a telecommunication network protocol. Its design and implementation may be based on the use of a telecommunication network protocol to provide character mapping and attention-handling capabilities. It also relies on the capability of a file transfer protocol to transfer files between host computers and data terminals. Remote job entry protocols are dependent on file-transfer (or data transfer) protocols, as discussed later in this chapter.

Telecommunication Network Protocols

Among all user-level protocols, telecommunication network protocols are the most well developed and best understood. The relatively successful designs and implementations of these protocols can be attributed mostly to the existence of a standard coded character set that enables interchange of information among computers and terminal devices. In networks such as the ARPANET where the end-to-end protocol provides a virtual line-switched environment for user process communication, a pair of simplex connections (or a full-duplex connection) is typically established by the initial connection protocol.[14] Data are then

[14] A. A. McKenzie, "Host/Host Protocols for ARPA Network," *Current Network Protocols*, NIC 8248 (Network Information Center, Stanford Research Institute, 1972).

transmitted over this virtual circuit interspersed with telecommunication control commands.

Host computers in a network must be able to support the great variety of terminals available now and in the future without an extremely high burden. Typically, this is done with the network virtual terminal. The network virtual terminal is an imaginary device which represents a network-wide standard description of basic terminals used in the network.* All host computers and terminal handlers map their local device characteristics so that these devices appear to have the characteristics and conventions of the network virtual terminal.

The network virtual terminal obviously cannot encompass the capabilities of all user terminals. Thus, an option negotiation mechanism is provided so that this standard representation of data terminals does not limit the service provided by a network virtual terminal. This mechanism allows the user (and/or the serving host) to request an option, such as changes in echo mode, line width, vertical tab, and remote controlled transmission. (For example, in a network virtual terminal, echo does not normally transverse the network; however, an

---

*For example, in the ARPANET, the network virtual terminal is a bidirectional character device with a keyboard and printer in line-buffered mode and with seven-bit ASCII character sets in an eight-bit field. Codes with the high-order bit set are reserved for TELNET control functions. Control codes include standard items such as LF, CR, BS, etc. For some control functions that do not have standard representations, standard methods to invoke them are provided. (For example, IP is for interrupting the process to which the network virtual terminal is connected, and AO is for allowing the current process to run to completion but not to send its output to the user.) The control functions are interpreted as NOP if they are not implemented by the host computer. The network virtual terminal has an unspecified carriage width and page length. The telecommunication protocol also needs to provide procedures by which a terminal user regains control of a runaway process. (For example, when buffer overflow occurs, the terminal control program in the remote host computer must not discard characters from the network. Rather, it may refuse further input via end-to-end flow control; however, such action would prevent attention characters from coming through the network as well.) For this purpose, the TELNET "SYNCH" signal is provided. A SYNCH signal consists of a HOST/HOST protocol "Interrupt From Sender" command over the control link, followed by a TELNET command DATA MARK over the data link. Upon receiving an INS, buffers between the user terminal and the host computer are emptied. The terminal control program in the remote host switches to process network input for attention characters and switches back to normal mode when a matching DM is received.

echo option can be negotiated so that echoes on the printer are controlled by the remote process.) The receiving party may accept the request when able and willing. Thus, users with more sophisticated terminals may obtain more than minimal service.

The currently used telecommunication network protocols differ only in features that are of secondary importance to the user (for example, whether control characters of the protocol are interspersed with the data stream, which requires all input characters to be scanned, or whether protocol message transmission relies on end-to-end protocol flow control). With possible development of a common network command language,[15] telecommunication network protocols will provide sufficient support for user terminal access of remote resources.

File Transfer Protocols

The ability of moving source code, machine language files, and data between host computers and data terminals is essential to implementing the batch processing function in a network. If a network is viewed as a distributed computation system with a distributed file system, then data transfer, file access, and file transfer capabilities are also essential first steps toward such a goal. For these reasons, file transfer protocols are among the earliest protocols developed. In most computer networks, ad hoc file transfer and data transfer procedures have been developed to provide needed support. (For example, in the ARPANET, the file transfer protocol supports transfer of records of sequential text files and block transfer of binary files between the user and the serving host computers, or between two serving host computers, neither of which is his/her local host. File transfer protocol commands include retrieve, store, create, delete, append, list, rename, etc.) However, unlike the case of telecommunication network protocols, designs of file transfer protocols have not yet been tried and proven satisfactory. Explicit user intervention is invariably required even in cases where the source and target file systems are compatible. Moreover, there is still no procedure to support transfer of structured files.

There is difficulty with a heterogeneous network, even at the data transfer level. Since storage representations in two host computers may be different, it is often necessary to perform some transformation on

---

15 G. M. Schneider, "DSCL--A Data Specification and Conversion Language for Networks," *Proc. ACM SIGMOD Conference* (ACM, 1975), pp 139-148.

the data.* Presently, most file transfer protocols allow the user to specify a representation type which implicitly or explicitly defines a byte size for interpretation of received data. The capability of transforming data to a form more suitable to user process needs has been provided by the Data Reconfiguration Service and Data Specification Conversion Language.[16-18] With these services, data is first mapped into an immediate form (virtual network representation).

Transferring files between two computer systems is difficult for many reasons. Lack of standard access commands and pathname conventions accounts for most of these difficulties. An obvious solution to this problem is the introduction of the network virtual file concept, which defines standard network commands and pathname conventions. (For example, a file name may be defined as a locally recognizable string terminated by a blank.) The network virtual file system provides an immediate representation of the source and target files, and thus eliminates the necessity that the source file be identical to the target file. Studies and evaluations of alternative designs of network virtual file systems that are now available should provide insight toward proper standard representation of file systems.

Another difficulty in file transfer is the handling of access control, which defines users' access privileges both to the system and to that system's files. This problem is currently an active research area in both the computer network and distributed system community and for designers of large data-base management systems.

---

16 J. P. Fry, R. L. Frank, and E. A. Hershey III, "A Developmental Model for Data Translation," *Proc. 1972 ACM SIGFIDET Workshop. Data Description. Access and Control* (1972), pp 71-106.

17 A. G. Merten and J. P. Fry, "A Data Description Language Approach to File Translation," *Proc. 1974 ACM SIGMOD Workshop on Data Description. Access and Control*, Randall Rustin, ed. (ACM, 1974), pp 191-206.

18 G. M. Schneider, "DSCL--A Data Specification and Conversion Language for Networks," *Proc. ACM SIGMOD Conference* (ACM, 1975), pp 139-148.

*For example, ASCII coded characters have different storage representations in different systems. In PDP-10's, five 7-bit ASCII characters are stored left-adjusted, in a 36-bit word. In IBM 360's, ASCII characters are stored in 8-bit EDBDIC codes. Similarly, there is a representation problem when transmission is binary between host computers with different word sizes.

To a great extent, these difficulties in file transfer are encountered even in local systems. Some problems, however, are unique to geographically distributed networks. For example, due to the relatively low bandwidth of the data link and high tariff, it is often too time-consuming and costly to move an entire file. Therefore, a file access protocol which supports addressing, retrieving, and transferring data within a file becomes a necessity. Ad hoc file access protocols do exist, but satisfactory designs of such protocols can only be obtained when the problem of network virtual file system and access control is solved.

Remote Job Entry Protocols

A remote job entry protocol specifies the procedure by which a user at one location causes a batch processing job to run at another location. It specifies the manner in which a user communicates via the network with a remote batch processing system, causing the system to retrieve the input file, process the job, and deliver the output file. Thus, if some mechanism for smooth, reliable file transfer without elaborate user intervention is used, many of the problems encountered in providing remote job entry capability become relatively simple. Many of these problems have been addressed and solved in existing and proposed remote job entry protocols.[19]

Future Developments in User Access Protocols

It is clear that many problems and unexplored areas remain in the development of user-level protocols, particularly in the areas of achieving true local-sharing capabilities within a network, maintaining distributed data bases (especially regarding security and reliability), coordination of cooperating distributed processes, and real-time communication. These problems have often been approached from the standpoint of rather narrow special interests. It appears that users of emerging networks are heading toward making the same mistakes that early programming language designers made--developing special-purpose, inflexible schemes for accomplishing tasks in what seems to be the most efficient way, and then becoming locked into those methods. Even when sophisticated user-level protocols emerge, there will be many users locked into their special-purpose mechanisms.

---

19 B. Bressler, R. Guida, and A. McKenzie, *Remote Job Entry Protocol*, NIC 12112 (Network Information Center, Stanford Research Institute, 1972).

A better approach at this time would be agreement between the major networks on standards for common "generic" functions. It would then be the responsibility of each server to provide a specified function, under an agreed-upon generic protocol. This really amounts to a revival of the UNCOL concept, which emerged in compiler technology after the proliferation of special-purpose languages. Currently, an initial attempt in this direction is the development of procedure call protocol, an interprocess or inter-host protocol that allows processes within one or more host computers in a network (specifically ARPANET) to communicate at a procedure call level.[20] The protocol is based on a model which views a process as a collection of remotely callable procedures. Each procedure can be invoked by name, and the model permits the process at either end of the inter-process communication channel to invoke procedures in its neighbor; in addition, the model permits a process to accept two or more procedure calls for concurrent execution. In effect, it makes the procedures of remote software systems accessible to the local programmer. The procedure call protocol defines the virtual programming environment in which remote procedures may be assumed to operate and in which the interprocess control exchange may be required to implement the virtual programming environment.

Other user protocols include the executive package, which includes procedures and data stores for user identification, accounting, and information; the file package, which specifies the procedures for opening, closing, and listing directories for creating, deleting, and renaming files, and for transfer files and file elements between processes; the batch job package which includes procedures for creating and deleting batch jobs, obtaining states of batch jobs, and communicating with operators of a batch processing host; and the low-level debug package, which includes procedures for a remote process of debugging at the assembly language level any process known to the local process.

The major criticisms of procedure call protocol's design include:[21]

1. Recovery from component malfunction may be awkward when handled by a process which is being manipulated by having its procedures executed.

2. The procedure call protocol seems too complicated to be used for processing which requires periodic but short interactions, and is probably too complex for implementation in a small host.

---

20 R. Schantz, *A Commentary on Procedure Calling as a Network Protocol*, RFC 684, NIC 32252 (Network Working Group, 1975).

21 *A Commentary on Procedure Calling as a Network Protocol*.

3. It forces a master/slave (superior/inferior) relationship that is unnatural for distributed systems.

4. The procedure call protocol formalism corrupts the clear notions of fork and join, since dummy returns (temporary disconnection instead of sharing of the communication link) may be required for long-delay activities.

5. It neither directly nor accurately models the network environment.

Further analysis and evaluation may prove that this protocol is unsuitable for widespread use and that using its primitives for the construction of distributed software systems may be too difficult. Nevertheless, its contribution will be in providing general principles which will aid in the future development of user-level protocols.

CITED REFERENCES

Bressler, B., R. Guida, and A. McKenzie, *Remote Job Entry Protocol*, NIC 12112 (Networks Information Center, Stanford Research Institute, 1972).

Burton, H. O. and D. O. Sullivan, "Errors and Error Control," *Proceedings of IEEE*, 60 (November 1972).

Cerf, V. G., *An Assessment of ARPANET Protocols*, RFC 635, NIC 30469 (Network Information Center, Stanford Research Institute, 1974).

Cerf, V. G. and R. E. Kahn, "A Protocol for Packet Network Intercommunication," *IEEE Trans Communications* (1974), pp 637-648.

Cerf, V. G., A. McKenzie, R. Scantleburg, and H. Zimmerman, "Proposal for an International End-to-End Protocol," *Computer Comm. Review*, Vol 6, No. 1 (1976).

Crocker, S. D., J. Heafner, R. Metcalfe, J. Postel, "Function-Oriented Protocols for ARPA Computer Network," *AFIPS 1972 SJCC Proc.*, 40 (1972), pp 271-279.

Fry, J. P., R. L. Frank, and E. A. Hershey III, "A Developmental Model for Data Translation," *Proc. 1972 ACM SIGFIDET Workshop, Data Description, Access and Control* (1972), pp 71-106.

Hovey, R. B., "Packet-Switched Network Agreed on Standard Interface," (1976).

Itzkowitz, A., *A Survey of the Properties of Computer Communication Protocols--Volume I, "The Function, Properties, Specifications, and Analysis Methods of Computer Communication Protocols*, Technical Report 0-1 (Construction Engineering Research Laboratory, June 1978).

Kleinrock, L., W. E. Naylor, and H. Opderbeck, "A Study of Line Overhead in ARPANET," *Comm. of ACM* (1976), pp 3-12.

McKenzie, A. A., "Host/Host Protocols for ARPA Network," *Current Network Protocols*, NIC 8248 (Network Information Center, Stanford Research Institute, 1972).

Merten, A. G. and J. P. Fry, "A Data Description Language Approach to File Translation," *Proc. 1974 ACM SIGMOD Workshop on Data Description, Access and Control*, Randall Rusing, ed. (Association for Computer Machinery [FACM], 1974), pp 191-206.

Schantz, R., *A Commentary on Procedure Calling as a Network Protocol*, RFC 684, NIC 32252 (Network Working Group, 1975).

Schneider, G. M., "DSCL--A Data Specification and Conversion Language for Networks," *Proc. ACM SIGMOD Conference* (ACM, 1975), pp 139-148.

UNCITED REFERENCES

Abramson, N., *ARPANET Satellite System, ASS Note 2,* NIC 11288 (Network Information Center, Stanford Research Institute, 1972).

"The ALOHA System," *Computer Communication Network,* Norman Abramson and Franklin Kuo, eds. (Prentice Hall, 1973).

Anderson, R. H., V. Cerf, E. F. Harslem, J. F. Heafner, J. Madden, B. Metcalfe, A. A. Shoshani, J. White, and D. Wood, "The Data Reconfiguration Service--An Experiment in Adaptable Process/Process Communication," *Proc. 2nd Symp. on Problems in the Optimization of Data Communication Systems,* P. E. Jackson, ed. (ACM, 1971).

Aupperle, E. M., *The Merit Network Re-examined,* MCN-0273-TP13 (Merit Computer Network, University of Michigan, 1973).

Belford, G. G., S. R. Bunch, J. Day, P. A. Alsberg, D. S. Brown, E. Grapa, D. C. Healy, J. R. Mullen, *A State of the Art Report in Network Data Management and Related Technology*, CAC Document No. 150 (Center for Advanced Computation, University of Illinois, 1975).

Benoit, J. W., "Evolution of Network User Services--The Network Resource Manager," *Proc. 1973 NBS IEEE Symposium on Computer Networks Trends and Applications* (IEEE, 1974), pp 21-24.

Bhushan, A., *The File Transfer Protocol,* MIT Project MAC (Network Working Group, 1972).

Carr, S., S. Crocker, and V. Cerf, "HOST/HOST Communication Protocols in the ARPA Network," *AFIPS 1970, SJCC Proc.,* 36 (1970), pp 859-597.

Crowther, W., "A System for Broadcast Communication: Reservation--ALOHA," *Proc. of 6th Hawaii International Conference on System Science* (1973).

DATAPAC, *Datapack Standard Network Access Protocol* (Computer Communication Group of Trans-Canada Telephone System, November 30, 1974).

Forgie, J. W., "Speech Transmission on Packet-Switched Store-and-Forward Networks," *Proc. AFIPS 1975 National Computer Conference,* 44 (1975), pp 137-142.

Fralic, S. C. and J. C. Garrett, "Technological Considerations for Packet Radio Networks," *Proc. AFIPS 1975 National Computer Conference,* 44 (AFIPS, 1975), pp 233-244.

Fralic, S. C., D. H. Brandin, F. F. Kuo, and C. Harrison, "Digital Terminals for Packet Broadcasting," *AFIPS Conference Proceedings,* National Computer Conference, Vol 44 (May 1975).

Kanodia, R. K., *A Lost Message Detection and Recovery Protocol,* RFC 663, NIC 31387 (Network Information Center, Stanford Research Institute, 1974).

Kimbleton, S. R. and G. M. Schneider, "Computer Communication Network: Approaches, Objectives and Performance Considerations," *Computing Surveys,* Vol 7, No. 3 (1976), pp 129-174.

Labonte, R. C., "A General Purpose Digital Communication System for Operation on a Conventional CATV System," *IEEE Comp. Conf., 1973* (1973), pp 85-88.

Martel, C. C., I. M. Cunningham, and M. S. Grushcow, "The BNR Network: A Canadian Experience with Packet Switching Technology," *Computer Network,* 2 (IFIPS, North-Holland Publishing Company, Amsterdam, 1974).

McKenzie, A. A., "TELNET Protocol Specification," *Current Network Protocols,* NIC 18639, NIC 18638-Revisions (Network Information Center, 1973).

The Merit Computer Network, Progress Report for the Period July 1969 March 1971," Publication 0571-PR-4 (Merit Computer Project, University of Michigan, May 1971).

Michener, J., I. Cotton, K. Keilley, D. Liddle, and E. Meyer, *Graphic Protocols,* NIC 15358 (Network Working Group, 1973).

Mimno, N. W., B. P. Cosell, D. C. Walden, S. C. Butterfield, and J. B. Levin,"Terminal Access to the ARPA Network: Experience and Improvements," *IEEE Compcon* (1973), pp 39-43.

Neumann, A. J., *User Procedures Standardization for Network Access,* Technical Note 799 (Systems Development Division, Institute for Computer Sciences and Technology, National Bureau of Standards, October 1973).

Padlipsky, M., *A Proposed Protocol for Connecting Host Computers to ARPA-like Networks via Front end Processors,* RFC 672, NIC 31117 (Network Working Group, October 1974).

Paoletti, L. M., "Autodin," *Computer Communication Networks* (Noerdhoff International Publ. Leyden, The Netherlands, 1975), pp 345-372.

Peterson, J. J. and S. A. Veit, *Survey of Computer Networks*, AD-763 068 (Mitre Corporation, September 1971).

Pouzin, L., "Presentation and Major Design Aspects of the CYCLADES Computer Network," *Data Networks--Analysis and Design*, Third Data Comm. Symp. (1973), pp 80-87.

Pouzin, L., "The Case for a Revision of X.25," *Computer Comm. Review*, Vol 6, No. 3 (1976), pp 17-20.

Protocol Specifications, NIC 18639 (Telnet, August 1973).

Pyke, T., Jr., "Computer Networking Technology--A State of the Art Review," *IEEE Computer*, Vol 6, No. 8 (1973), pp 12-19.

Roberts, L. G. and B. D. Wessler, "Computer Network Development to Achieve Resource Sharing," *Proc. of SJCC, AFIPS* (1970), pp 543-549.

Roberts, L. G., "Extensions of Packet Communication Technology to a Hand Held Personal Terminal," *Proc. of the SJCC* (AFIPS, 1972), pp 295-298.

Stefferud, E. D., L. Grobstein, and R. Uhlig, "Wholesale/Retail Specialization in Resource Sharing Networks," *IEEE Computer*, 6, 8 (August 1973), pp 31-37.

Teshigawara, R. H., "Computer Networks in Japan," *Computer Comm. Review*, Vol 6, No. 3 (1976), pp 4-13.

Thomas, R. H. and D. A. Henderson, "McRoss--A Multi-Computer Programming System," *Proc. AFIPS 1972 Spring Jt. Computer Conf.*, 42 (AFIPS Press, 1972), pp 281-293.

Thomas, R. H., "On the Design of a Resource Sharing Executive for the ARPANET," *Proc. AFIPS 1972 National Computer Conf.*, 42, (AFIPS Press, 1973), pp 155-164.

Tymes, LaRoy, "Tymnett--A Terminal-Oriented Communication Network," *AFIPS 1972 Spring Jt. Computer Conf.*, 38 (AFIPS Press, 1971) pp 211-216.

Vallee, J., H. M. Lipinski, and R. H. Miller, *Group Communication through Computers, Vol. 1, Design and Use of the Forum System*, Report R-32, (Institute for the Future, July 1974).

Walden, D., "A System for Interprocess Communication in a Resource Sharing Computer Network," *Comm. of ACM* (1974), pp 221-230.

Watson, R. W., *Some Factors Which a Network Graphic Protocol Must Consider*, RFC 192, NIC 7137 (Network Working Group, 1971).

White, J. E., "The Procedure Call Protocol," *NIC 29459* (Network Information Center, Stanford Research Institute, 1975).

Zafiropula, P., E. Port, and K. Kummerle, *Extension of a Circuit Switched User Network Interface to Packet Switching*, Technical Report Z758, (IBM, Thomas J. Watson Research Center, 1976).

Zimmermann, H., and M. Elie, *Transport Protocol, Standard Host-Host Protocol for Heterogeneous Computer Networks*, IFIP W. G. 6.1 General Note 61 (IFIP, 1975).